**Software Architecture Document**

**Версия 0.2**

**История изменений**

|  |  |  |  |
| --- | --- | --- | --- |
| **Дата** | **Версия** | **Описания** | **Автор** |
| 22.04.2016 | 0.1 | Описание архитектуры |  |
| 11.05.2016 | 0.2 | Диаграммы |  |

## Назначение

Целью данного документа является подробное архитектурное описание разрабатываемой программной системы. В нём приводится самые значимые архитектурные решения, описание отдельных компонентов, а также принцип взаимодействия между ними. Этот документ используется командой разработчиков в качестве основного документа при реализации системы.

**Сфера применения**

Документ разрабатывается в рамках этапа проектирования программной системы. Он предназначен для описания архитектуры, свойств и связей программного продукта «Система контроля расписания поездов» СКОРП и в дальнейшем будет использоваться разработчиками, для написания программного кода.

## Определения и аббревиатуры

Оператор РЖД – работник РЖД или программная система, осуществляющий(ая) контроль над транспортными перевозками по сети железных дорог, а также обеспечивающий(ая) связь между отдельными узлами этой сети.

Клиент РЖД – пользователь услуг, предоставляемых РЖД (транспортные перевозки и информация о движении поездов).

Системный администратор РЖД – работник РЖД, ответственный за управление клиентскими базами данных, настройка и последующее сопровождение компьютерных сетей РЖД.

ЖД – набор станций соединённых путями, по которым могут перемещаться поезда.

Железнодорожная станция – это пункт отправки или прибытия поездов, для которого характерны следующие черты: наличие нескольких транспортных линий; присутствие максимум одного поезда на линии в один момент времени; наличие у станции заранее известного расписания, которое не меняется как минимум в течение одного рабочего дня.

РЖД (“Российские железные дороги”) – российская государственная компания, предоставляющая услуги по использованию сети железных дорог. Полное наименование — Открытое акционерное общество “Российские железные дороги”.

## 

## Ссылки

1. Глоссарий
2. Видение
3. Диаграмма пакетов
4. Диаграмма классов
5. Диаграмма компонентов
6. Диаграмма состояний
7. Диаграмма вариантов использования
8. Диаграмма последовательности
9. Диаграмма объектов

**Описание архитектуры системы**

При разработке программной системы за основу была взята модель MVC.

**Model-view-controller** (**MVC**) — схема использования нескольких шаблонов проектирования, с помощью которых модель приложения, пользовательский интерфейс и взаимодействие с пользователем разделены на три отдельных компонента таким образом, чтобы модификация одного из компонентов оказывала минимальное воздействие на остальные.

В соответсвии с этой моделью система разделена на три логических компонента:

GUI – Графический интерфейс с которым взаимодейсвует пользователь

Ядро – содержит в себе основную логику системы и контейнер

БД Менеджер – предназначен для взаимодействия с БД

## Отображение сценариев использования

Use Case Diagram – диаграмма использования. Диаграмма, отражающая отношения взаимодействия User и Operator и Administrator с системой.

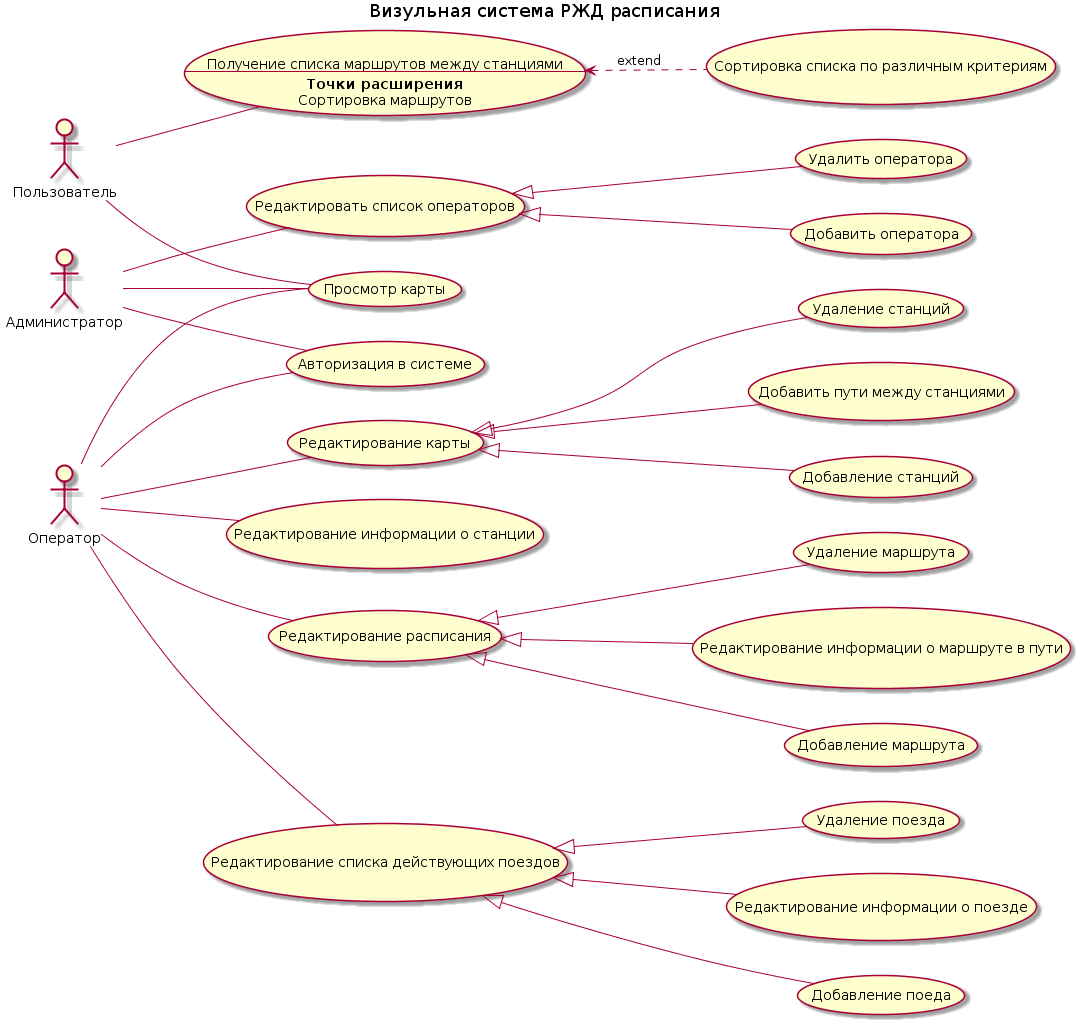
User: Цель – является клиентом РЖД его целью является найти оптимальный для маршрут из станции А в станцию Б.

Operator: Работник РЖД в задачи которого входит отслеживание и редактирование расписания поездов и карты маршрутов.

Administrator: проводит первоначальную настройку сети управляет аккаунтами операторов.

Use Case View Описывает набор сценариев и/или вариантов использования, которые представляют некоторую существенную, центральную функциональность. Он также описывает набор сценариев и/или вариантов использования, которые имеют существенное архитектурное освещение. Иллюстрирует конкретные нюансы архитектуры.

Use Case диаграмма:



Сценарии использования данной системы представлены ниже.

* Подобрать оптимальный маршрут из точки А в точку Б.
* Редактирование расписания.
* Редактирование карты.
* Редактирование списка действующих поездов
* Редактирование информации о станции
* Редактирование списка операторов

1. Сценарии использования

Описание вариантов использования

*Сценарий 1.*

**Название**: «Подобрать оптимальный маршрут из точки А в точку Б».

**Роль:** Клиент РЖД

**Начальное состояние**: приложение проинициализировано.

**Основной поток событий**:

1. Пользователь просматривает карты и выбирает себе точку отправки и точку прибытия(Точки А и Б)

2. Пользователь выбирает пункт меню «составить маршрут»

3. Вводит в поле пункт отправки точку А

4. Вводит в поле пункт назначение точку Б

5. Вводит в поле время отбытия удобное уме время

6. Нажимает кнопку составить маршрут

7. Пользователь получает оптимальные маршруты

*Сценарий 2.*

**Название**: «Редактирование расписания».

**Роль:** Оператор РЖД

**Начальное состояние** приложение проинициализировано

**Основной поток событий**:

1. Оператор авторизуется.

2. Оператор нажимает кнопку редактирования расписания

3. Редактирует расписание ( изменяет время прибытия и отправки на какую либо станцию).

4. Нажимает кнопку сохранить изменения

*Сценарий 3.*

**Название**: «Редактирование карты».

**Роль:** Оператор РЖД

**Начальное состояние** приложение проинициализировано

**Основной поток событий**:

1. Оператор авторизуется.

2. Оператор нажимает кнопку редактирования карты

3. Редактирует карту ( добавляя или удаляя какие либо станции).

4. Нажимает кнопку сохранить изменения

*Сценарий 4.*

**Название**: «Редактирование списка действующих поездов».

**Роль:** Оператор РЖД

**Начальное состояние** приложение проинициализировано

**Основной поток событий**:

1. Оператор авторизуется.

2. Оператор нажимает кнопку редактирования списка поездов

3. Редактирует список поездов ( добавляя или удаляя какие либо поезда из списка).

4. Нажимает кнопку сохранить изменения

*Сценарий 5.*

**Название**: «Редактирование информации о станции».

**Роль:** Оператор РЖД

**Начальное состояние** приложение проинициализировано

**Основной поток событий**:

1. Оператор авторизуется.

2. Оператор нажимает кнопку редактирования информации о станции

3. Редактирует информацию о станции (изменение названия).

4. Нажимает кнопку сохранить изменения

*Сценарий 6.*

**Название**: «Редактирование списка операторов».

**Роль:** Администратор

**Начальное состояние** приложение проинициализировано

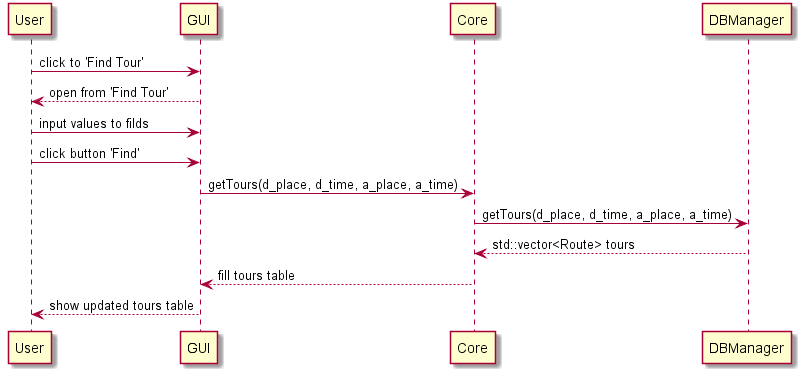
**Основной поток событий**:

1. Администратор авторизуется.

3. Администратор удаляет оператора или подтверждает его регистрацию

4. Нажимает кнопку сохранить изменения

Диаграмма последовательности:



## Логическое представление

### Обзор

Описание логической архитектуры. Описаны наиболее важные классы, их организация в пакеты и подсистемы, и организация этих подсистем. Также имеет место диаграммы классов, иллюстрирующая зависимости между архитектурно значимыми классами и подсистемами.

### 

Существующие подсистемы:

GUI – графический интерфейс

Ядро – основная логика системы, работающая с контейнером

БД менеджер – работает с БД

### UML диаграмма классов архитектуры всей системы

Диаграмма ядра:

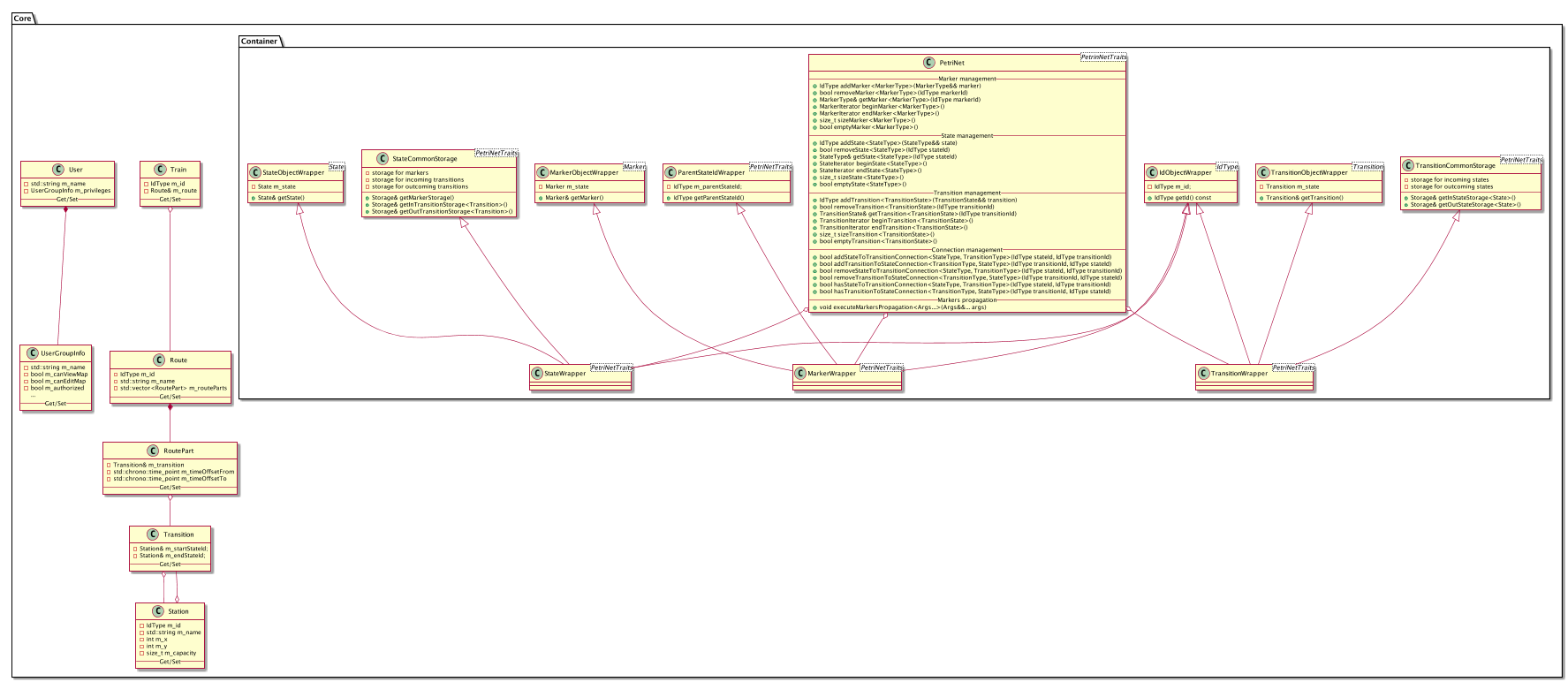


Диаграмма GUI:

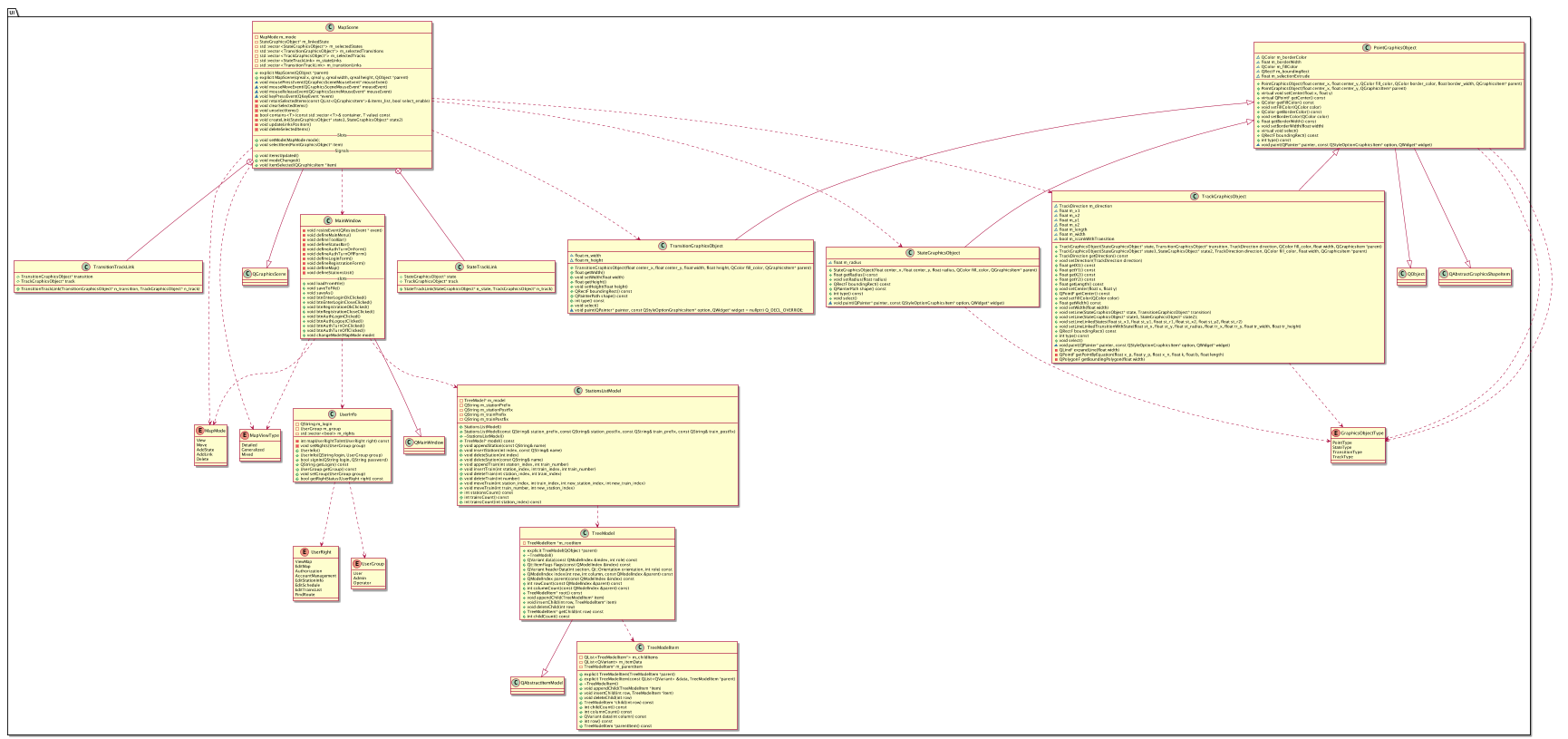


Диаграмма БД:



Диаграмма компонентов:

